ATLAS ANALYSIS PERFORMANCE ON THE GRID

monitoring and improving

Local analysis perf.

- Runs for 6 months already
- All T2Ds
- 8 test types
- Presented at US facilities meeting at Wisconsin (Ilija), TIM (Wahid)

WAN federation

- Runs for 2 months
- BNL T1 and all T2 sites + CERN
- Presented at TIM (Doug)

DPM dedicated

- New development
- DPM specific tests
- http, nfs4.1
- New xrootd clients

Framework

- HC submits jobs to ANALY queues
- Jobs are instrumented and send data to dedicated Oracle DB
- Dedicated web sites
- Same files: official D3PDS
- Same code: generic + realistic analysis

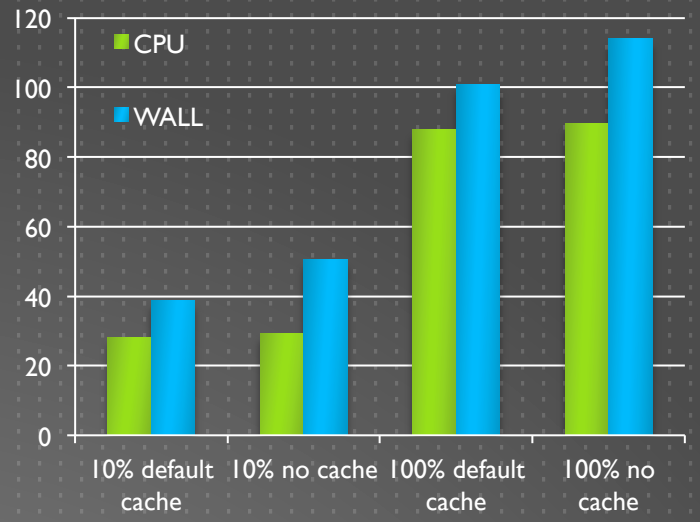
UNIVERSAL LOCAL IO TESTS

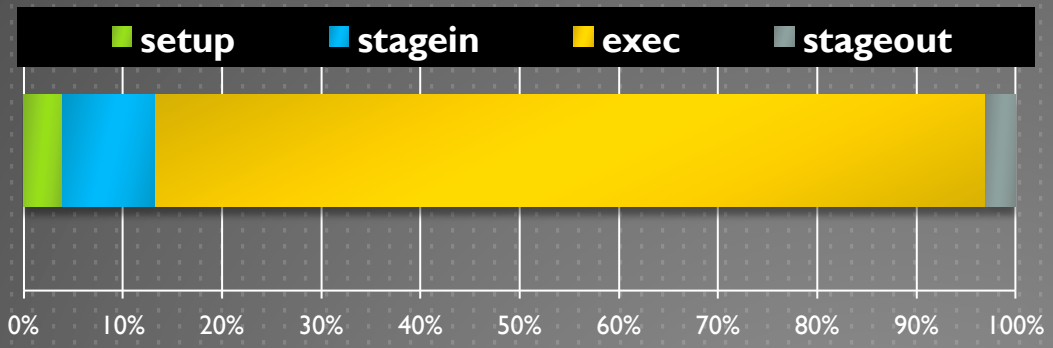
- Know what is performance of ATLAS jobs on the grid
  - We don't have one widely used framework that we could instrument so we need to be open to any kind of jobs: root analysis scripts, athena jobs, d3pd maker
- Understand the numbers we get
- Improve
  - Our software — In need of better BS optimization for ROOT
  - Our files — Some improvements already in (compression f. = 6)
  - Way we use root
  - Middleware
  - Sites

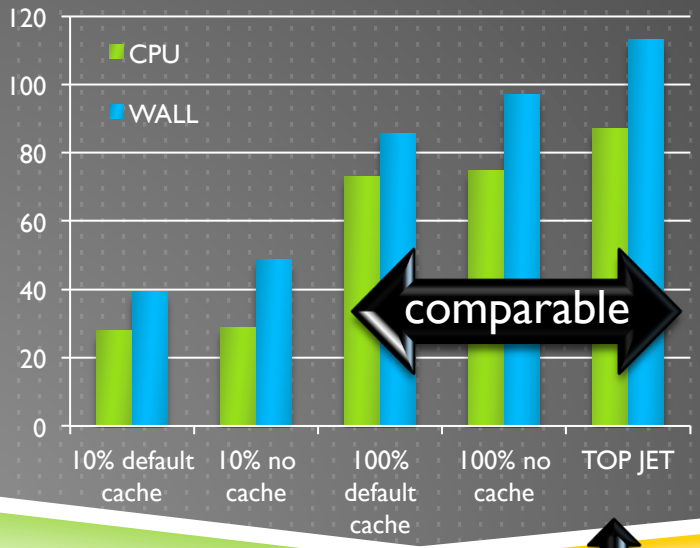
  Concentrate on these
- Way to test developments
- Have it as simple, realistic, accessible, versatile as possible
  - Running on most of the resources we have
  - Fast turn around
  - Test codes that are "recommended way to do it"
  - Web interface for most important indicators

TESTS CPU EFFICIENCY

- Average results over all the sites during February using 17.0.4 (ROOT 5.28)
- 77% Event loop CPU efficiency
- Total job CPU efficiency 41%

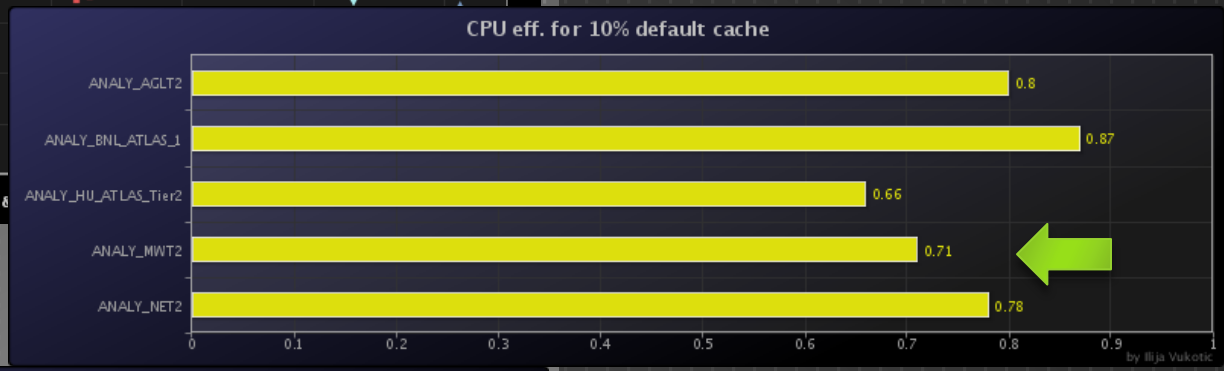


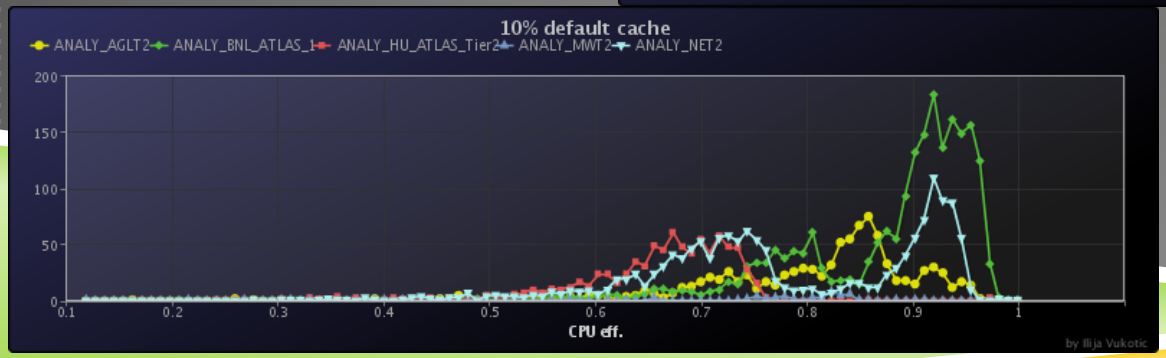




Realistic analysis

SOME PERFORMANCE PLOTS

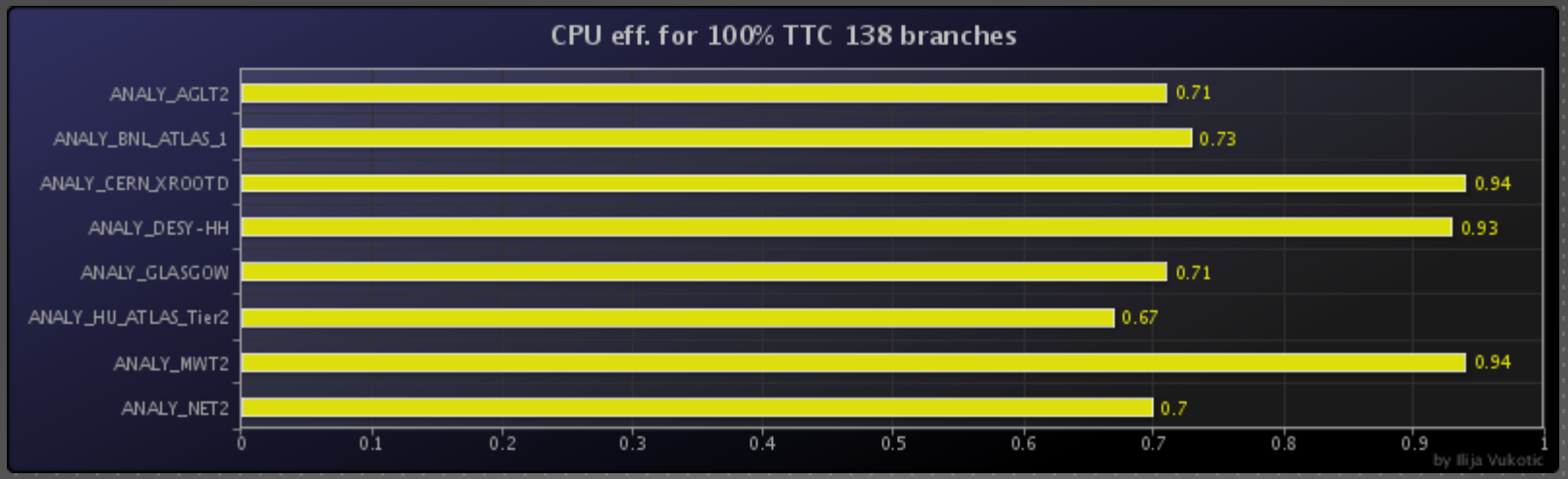




- Hundreds of jobs per day
- Under real world conditions
- Realistic and generic jobs

- Sites can not have the same performance due to different
  - Hardware
  - Percentage of analysis jobs
  - Storage
  - Configurations

SOME PERFORMANCE PLOTS



- Can compare to a lot of sites
- Try to learn from the best ones
- Optimize individually
  - for each site compare stage-to-tmp-disk vs. direct access
  - Optimal overbooking

TTC seems more effective on sites with good vector read (dCap++; xrootd) than when reading from local disk. Even not taking stage in time into account!

OPTIMIZING US SITES

- It's a big job to optimize all the sites. let's start with US ones
- Will need a full time interaction with sites
- First collect information on sites, similar to what was done for lustre and gpfs sites
  - https://docs.google.com/spreadsheet/ccc?key=0AjDZJgYDLICadFVFQkFFczdORDY2bC1raTRkd21hN1E
  - https://docs.google.com/spreadsheet/ccc?key=0AqcCwHr39RA6dGdiMU5aajNvYnNSRktoOWhSQ3V5aWc
- Try to understand what are the main factors influencing performance
  - Number of nodes, disk servers
  - Network topology
  - Metadata servers
  - Their configuration, read-ahead size
  - Loads on site (specially other VOs)
  - Software versions
- Improve, document and share knowledge

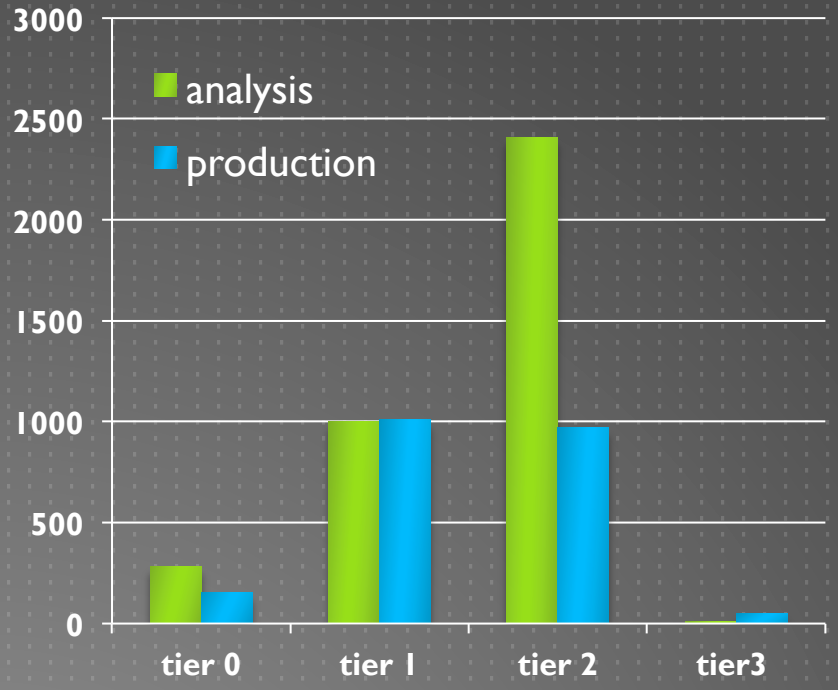
PROPOSAL

- All US Tier2s meet twice a month to discuss progress and share knowledge
- I can organize meetings and documentation
- Between meetings work one-on-one with sites.
- Report on progress at US ATLAS facilities integration meeting
- 6 months should be enough to bring sites to whatever is their optimum.

RESERVE

WHY ANALYSIS JOBS ARE IMPORTANT ?

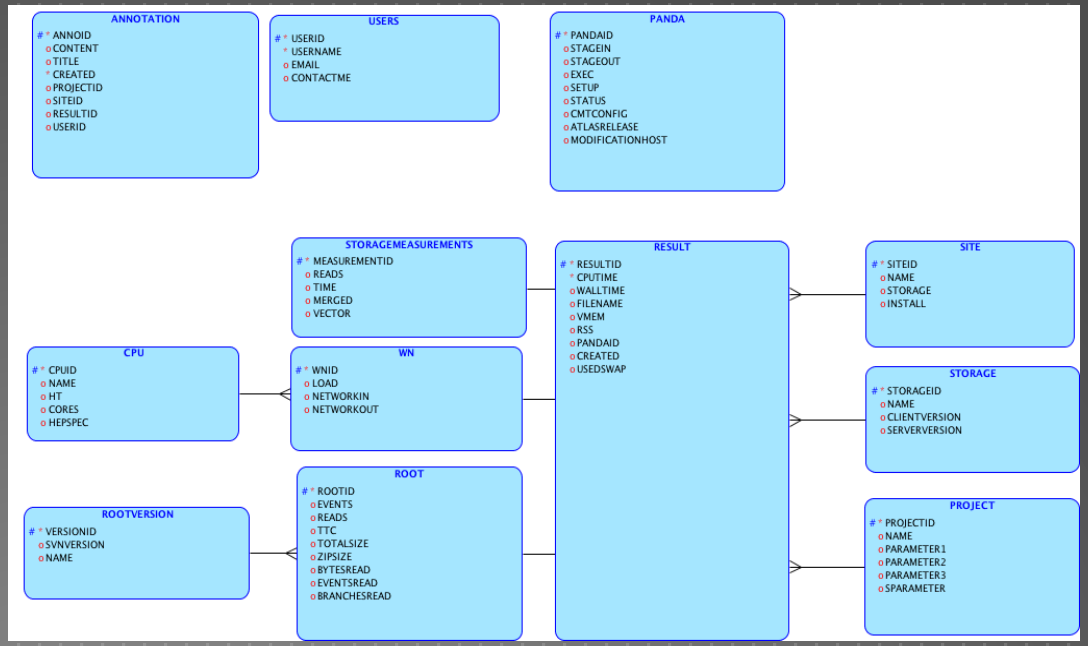
- Number of analysis jobs are increasing
- Production jobs are mostly CPU limited, well controlled, hopefully optimized and can be monitored through other already existing system
- Analysis jobs we know very little about and potentially could: be inefficient, wreck havoc at storage elements, networks.

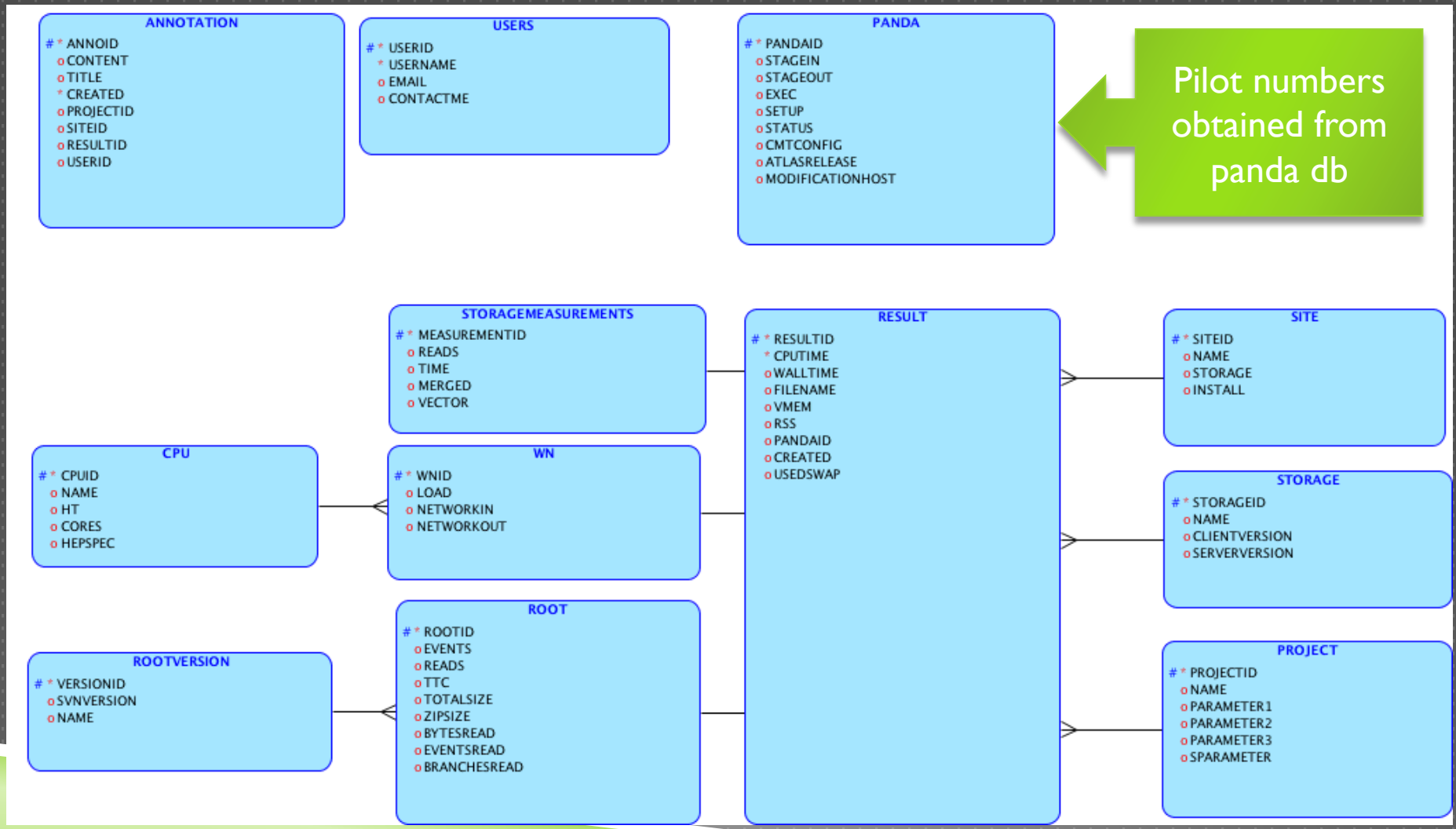


HOW ITS DONE

1. HammerCloud submits jobs
2. Jobs collects and sends info to DB

- Continuous
  - Job performance
    - Generic ROOT IO scripts
    - Realistic analysis jobs
  - Site performance
  - Site optimization
- One-off
  - new releases (Athena, ROOT)
  - new features, fixes
- All T2D sites (currently 35 sites)
- Large number of monitored parameters
- Central database
- Wide range of visualization tools



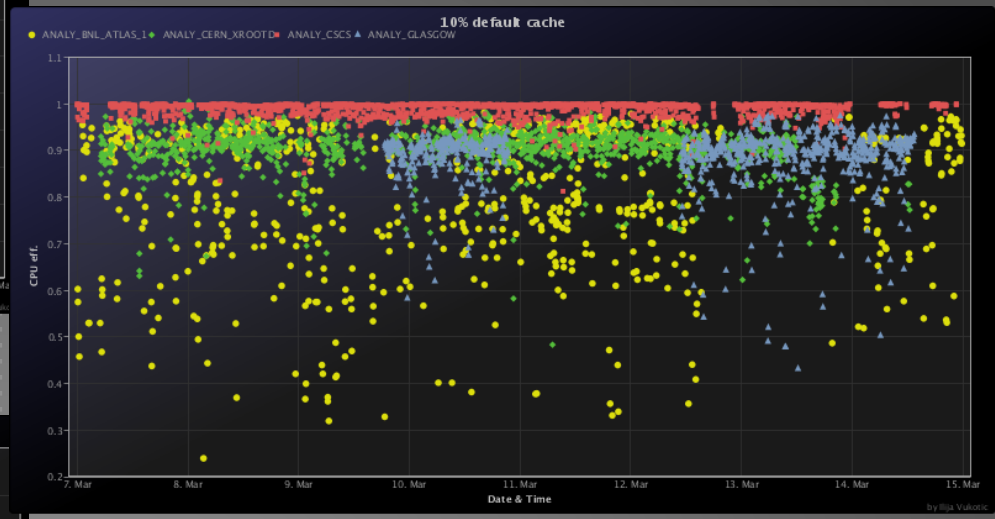

ANNOTATION
* ANNOID
o CONTENT
o TITLE
* CREATED
o PROJECTID
o SITEID
o RESULTID
o USERID
USERS
* USERID
* USERNAME
o EMAIL
o CONTACTME
PANDA
* PANDAID
o STAGEIN
o STAGEOUT
o EXEC
o SETUP
o STATUS
o CMTCONFIG
o ATLASRELEASE
o MODIFICATIONHOST
Pilot numbers obtained from panda db
STORAGEMEASUREMENTS
* MEASUREMENTID
o READS
o TIME
o MERGED
o VECTOR
RESULT
* RESULTID
* CPUTIME
o WALLTIME
o FILENAME
o VMEM
o RSS
o PANDAID
o CREATED
o USEDSWAP
SITE
* SITEID
o NAME
o STORAGE
o INSTALL
CPU
* CPUID
o NAME
o HT
o CORES
o HEPSPEC
WN
* WNID
o LOAD
o NETWORKIN
o NETWORKOUT
STORAGE
* STORAGEID
o NAME
o CLIENTVERSION
o SERVERVERSION
ROOT
* ROOTID
o EVENTS
o READS
o TTC
o TOTALSIZE
o ZIPSIZE
o BYTESREAD
o EVENTSREAD
o BRANCHESREAD
ROOTVERSION
* VERSIONID
o SVNVERSION
o NAME
PROJECT
* PROJECTID
o NAME
o PARAMETER1
o PARAMETER2
o PARAMETER3
o SPARAMETER

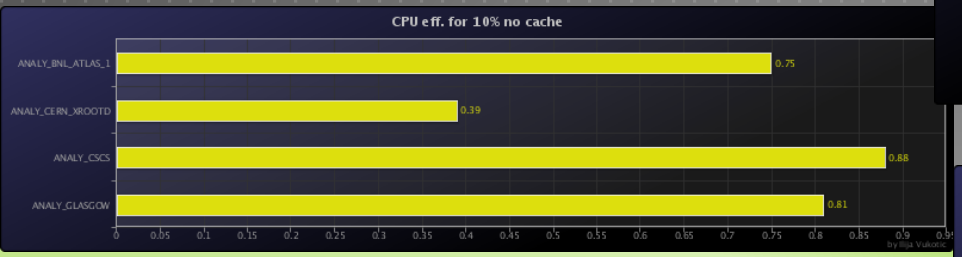
MESSAGE

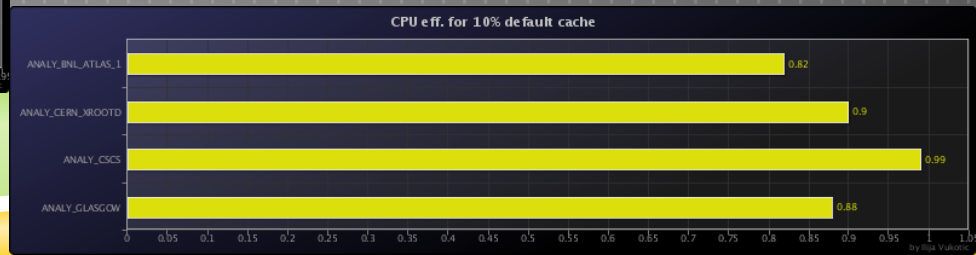
- Everybody
  - Visit http://ivukotic.web.cern.ch/ivukotic/HC/index.asp
  - Give it a spin, give us feedback and ask for features
- Site admins
  - We are trying to improve our performance and reduce stress on your systems, and not to judge sites.
  - Compare your site to others, see what they do differently and improve.
- ROOT / CMS / StorageTesting people
  - Give us you code/data and we do fast testing for you on all different kinds of CPUs / storage backends / protocols.
  - We'll learn something from your tests too.

RESULT – EFFICIENCY OF TTC

- For EOS it is indispensable

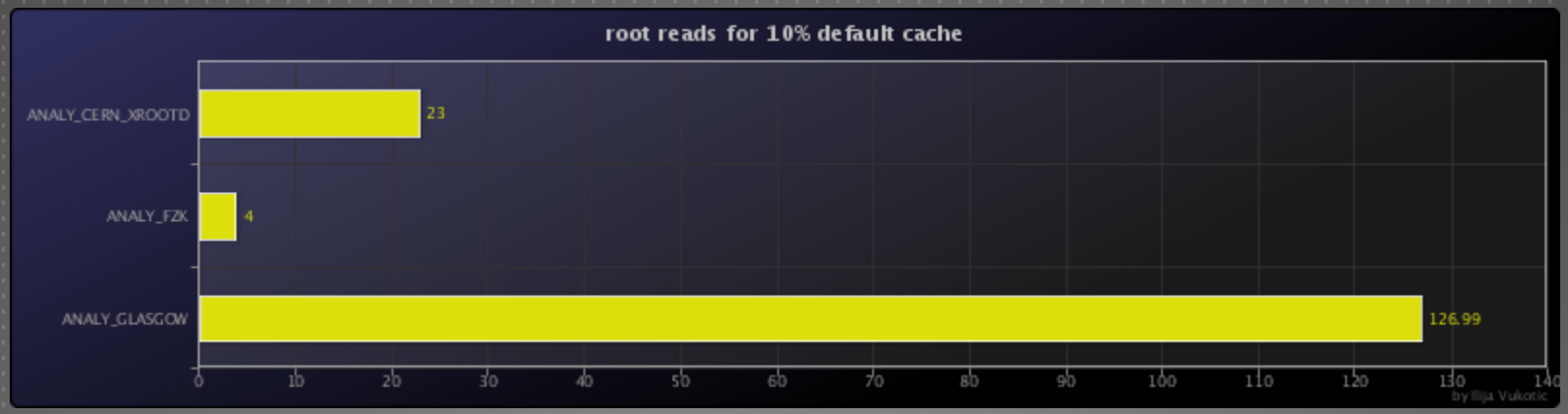






RESULT – EFFICIENCY OF TTC

- TTC effects will get more pronounced over WAN



RESULT – SETUP TIME PART I

Even under one minute the setup time is way too large overhead for analysis jobs. Analysis jobs duration limited by size of temp disk (<10GB).
Any reasonable analysis job should be shorter than 20 min.

At some sites we occasionally noticed very large setup times.

- They allow for 24 jobs per machine and these machines have 24GB of RAM,
- To avoid swapping problems they make accepted job wait in setup until there is 2GB of RAM free.
- Occasionally this leads to job waiting hour or two in setup .
- Even then the job often runs into swapping problem few minutes later.

At some CVMFS sites setup times in thousands of seconds traced to a bug in CVMFS that causes cache corruption.

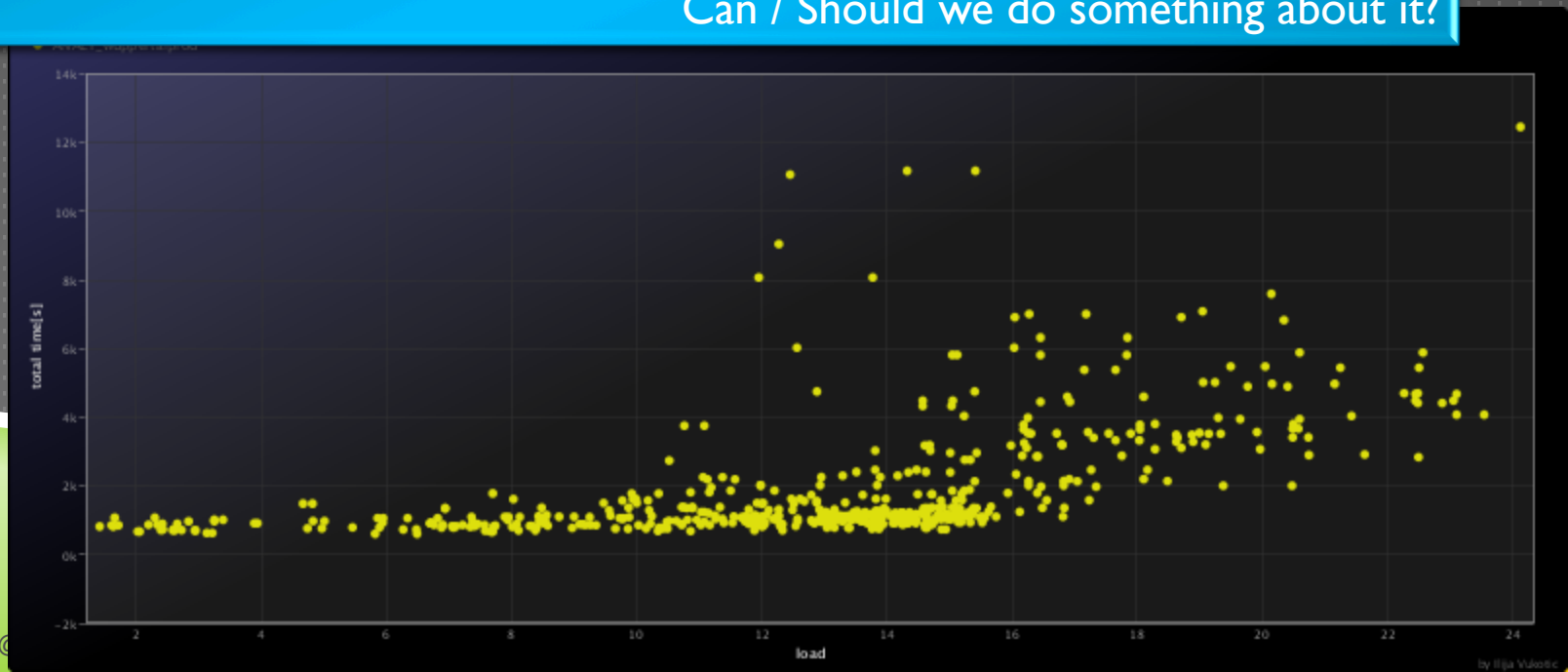
The biggest problem are times of 50-100 seconds. Against all the expectations CVMFS sites are in average slower to setup: 40 vs 52 seconds

- Is cache invalidated so often?
- Very big and a long standing issue of CMT doing millions of stat calls.
- Working on it with David Q., Grigori R.

RESULT - OVERBOOKING

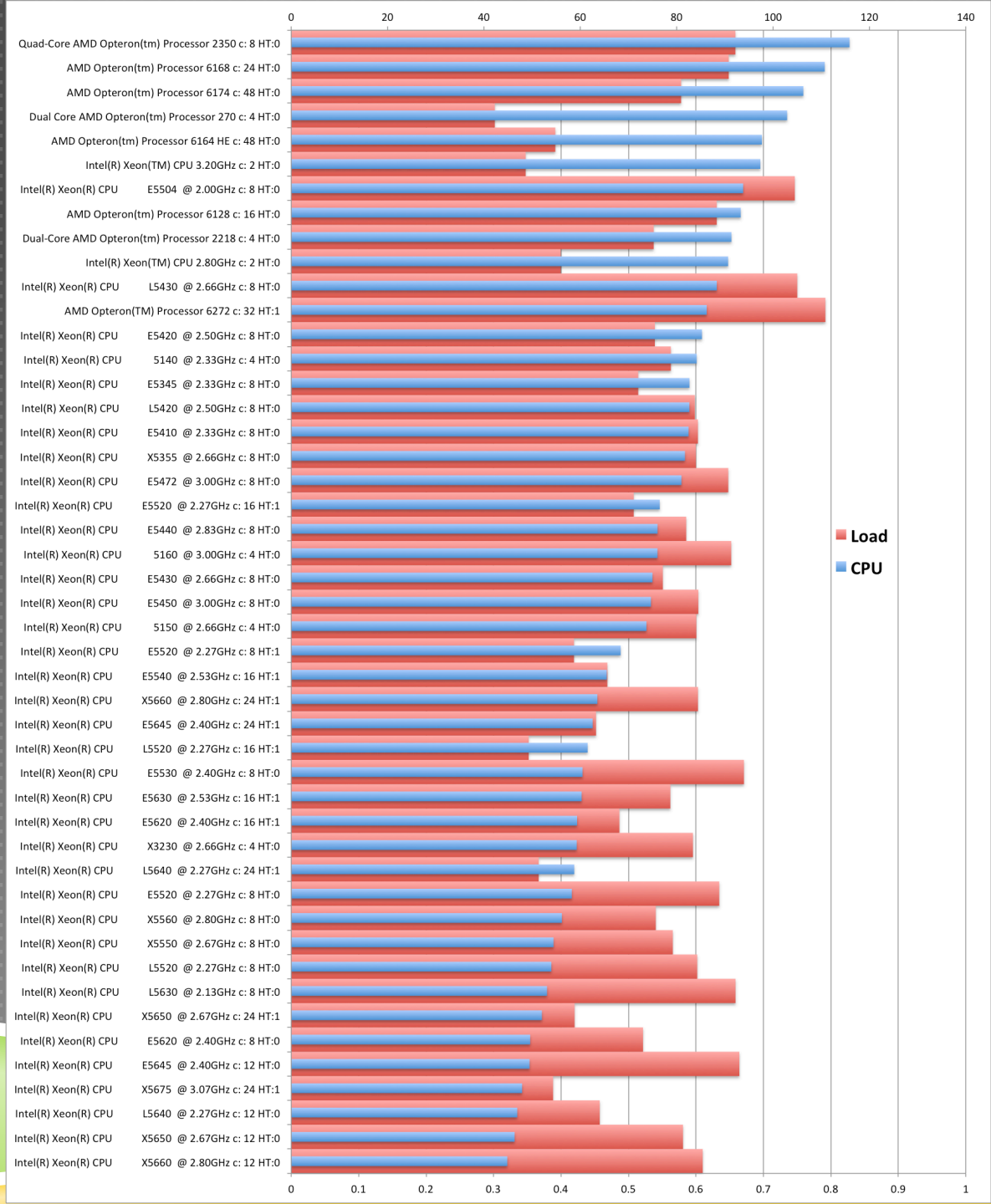
- There is often a suboptimal overbooking of the nodes.
- Example
  - use Intel(R) Xeon(R) CPU E5645 @ 2.40GHz, 12 cores machines.
  - While loads up to 14 -15 are maybe acceptable loads of 16+ are just wasting resources as job execution times basically doubles.

There is nothing preventing any grid job spawning 15 threads.
This affects everybody.
Can / Should we do something about it?



CPU NORMALIZATION

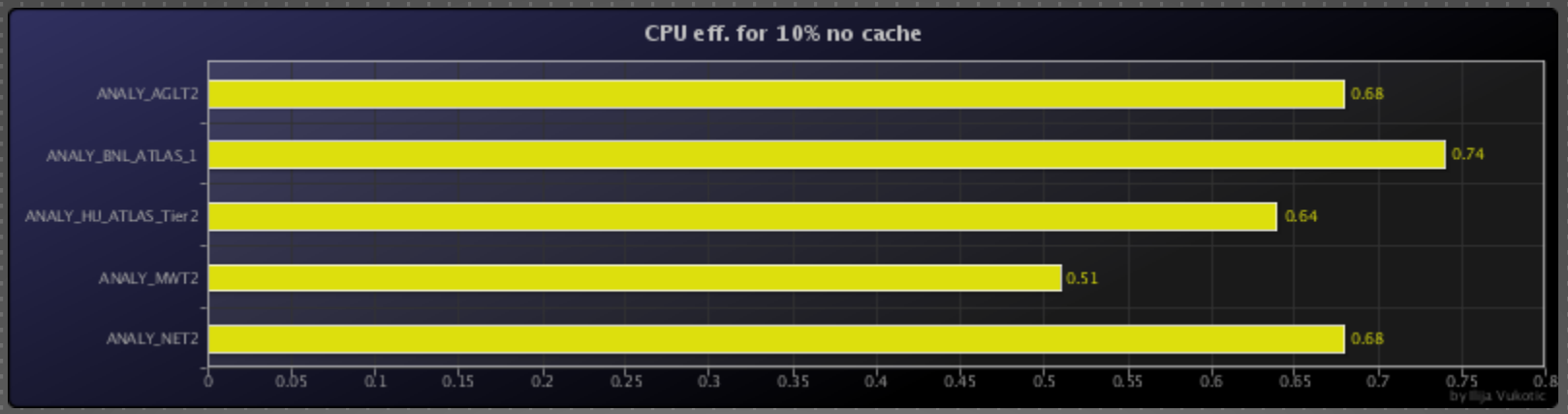
- CPU HS06 not a reliable indicator of how much CPU time our jobs will spend
- Use our jobs to derive this info

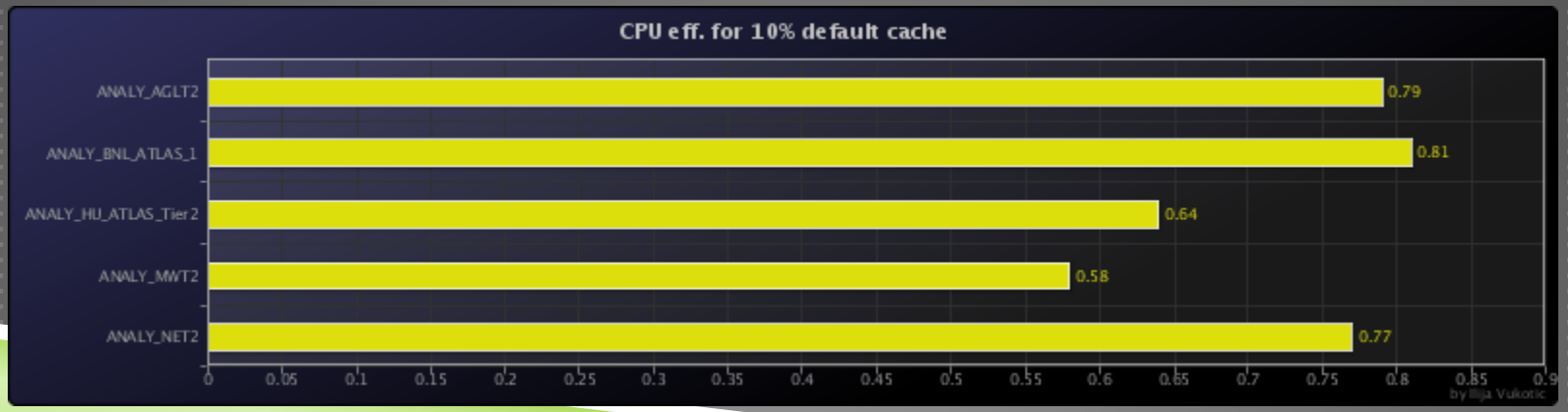


ILIJA VUKOTIC IVUKOTIC@CERN.CH

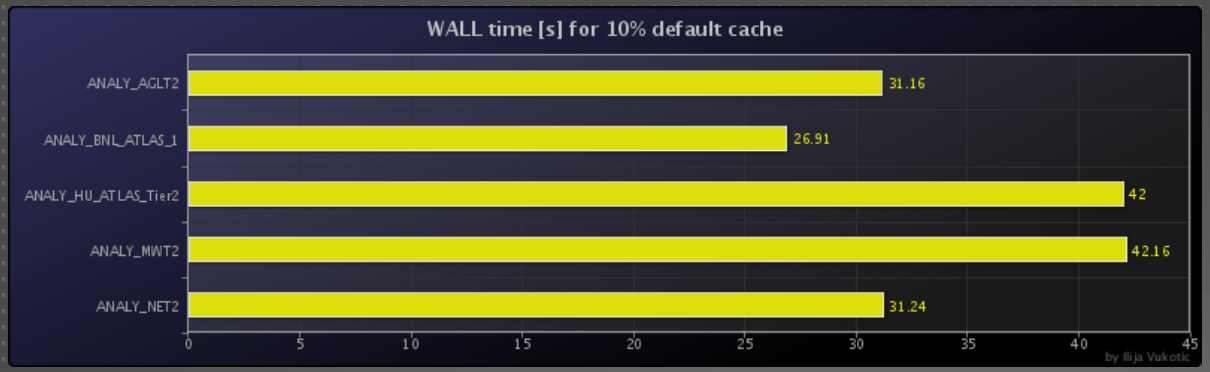
US sites

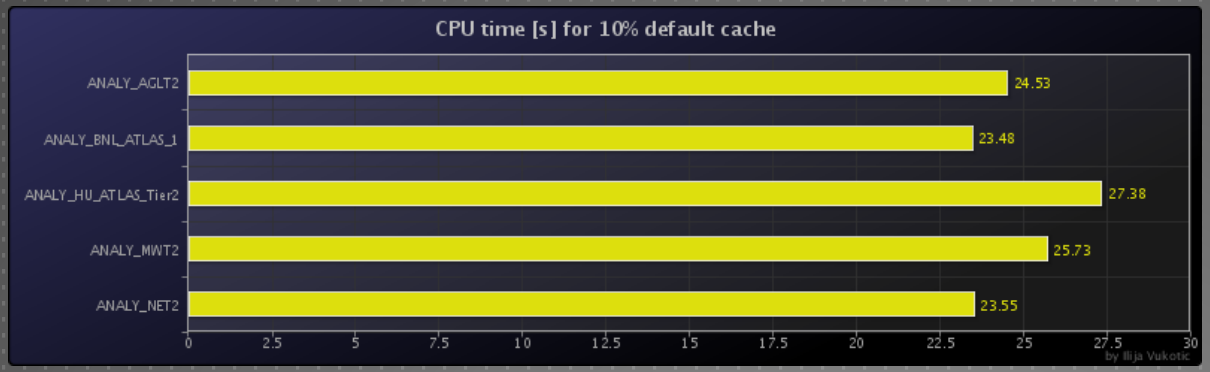
EFFICIENCY - OLD

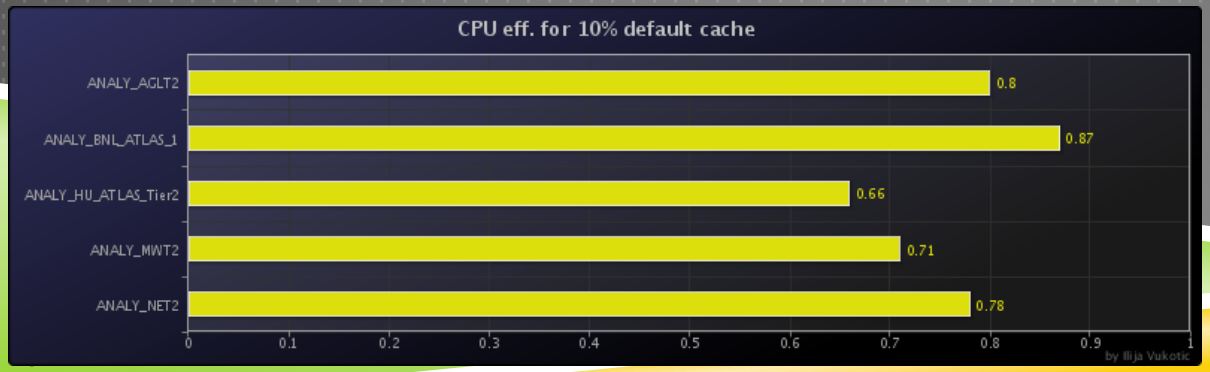




SOME PERFORMANCE PLOTS - NEW



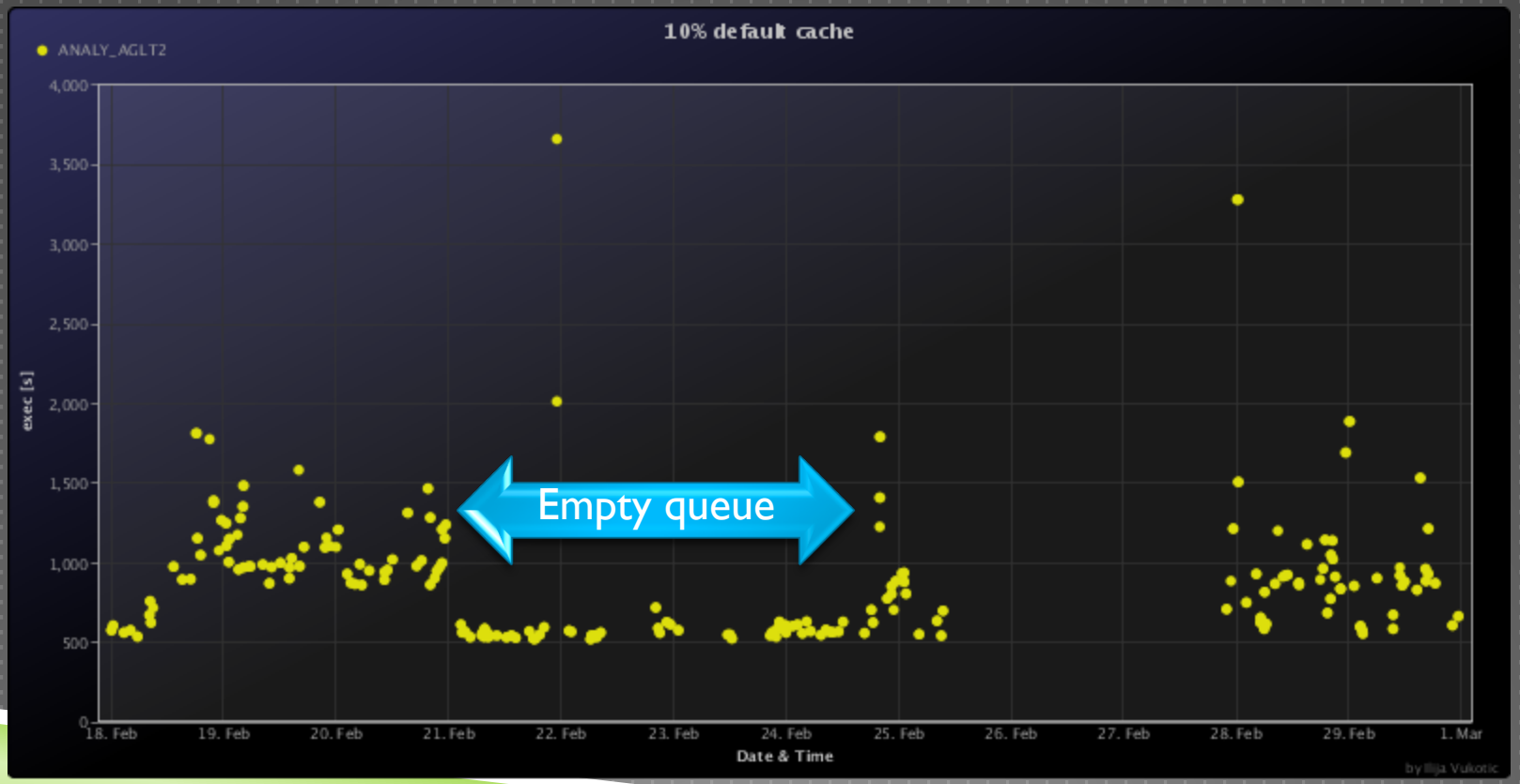




US sites

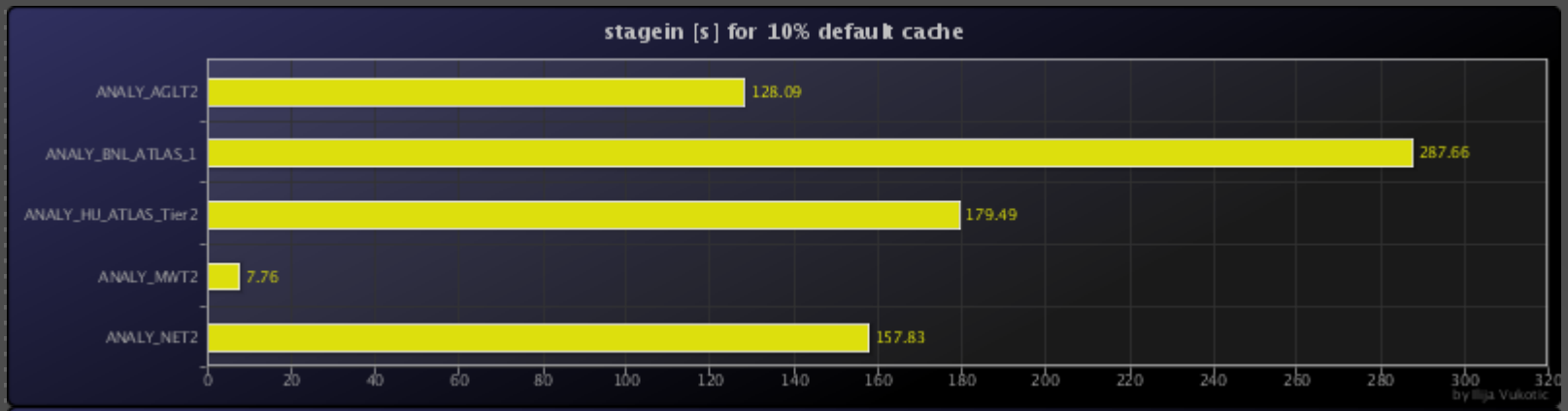
EFFICIENCY

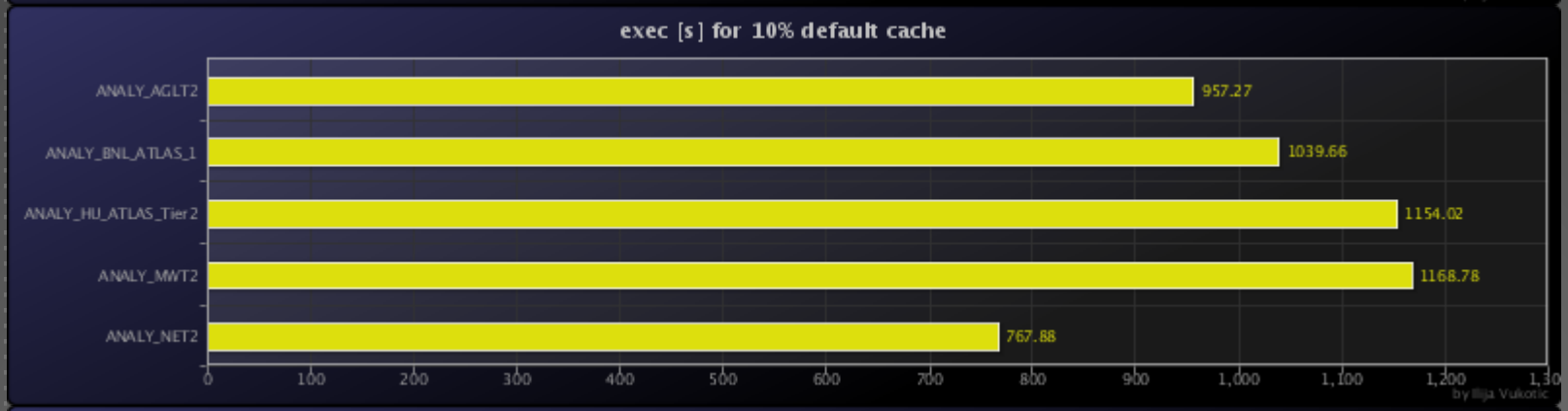
WN load is not very correlated to CPU eff. But site occupancy may be.

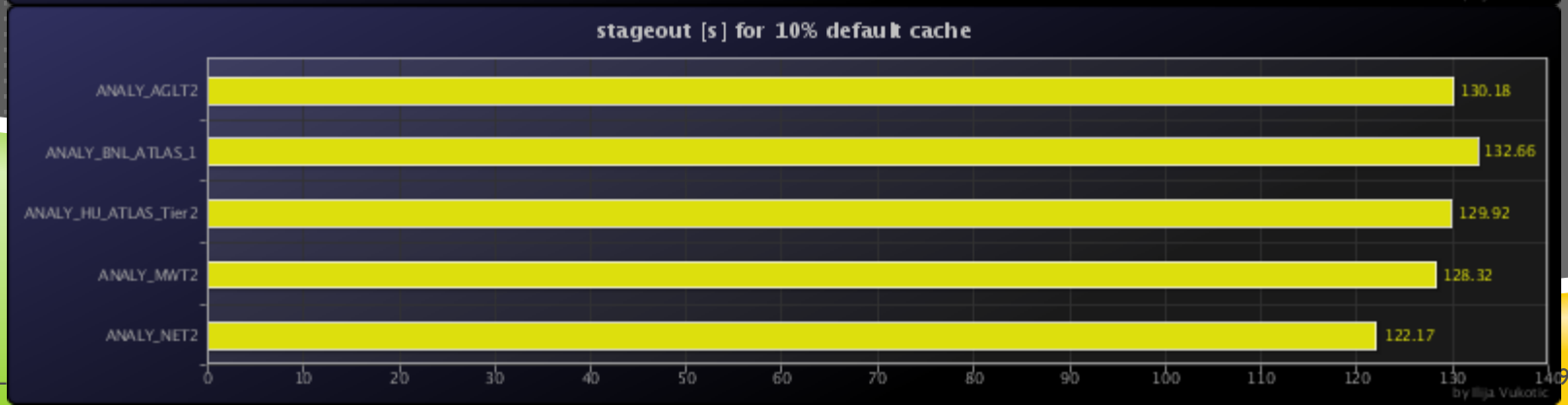


PILOT TIMINGS

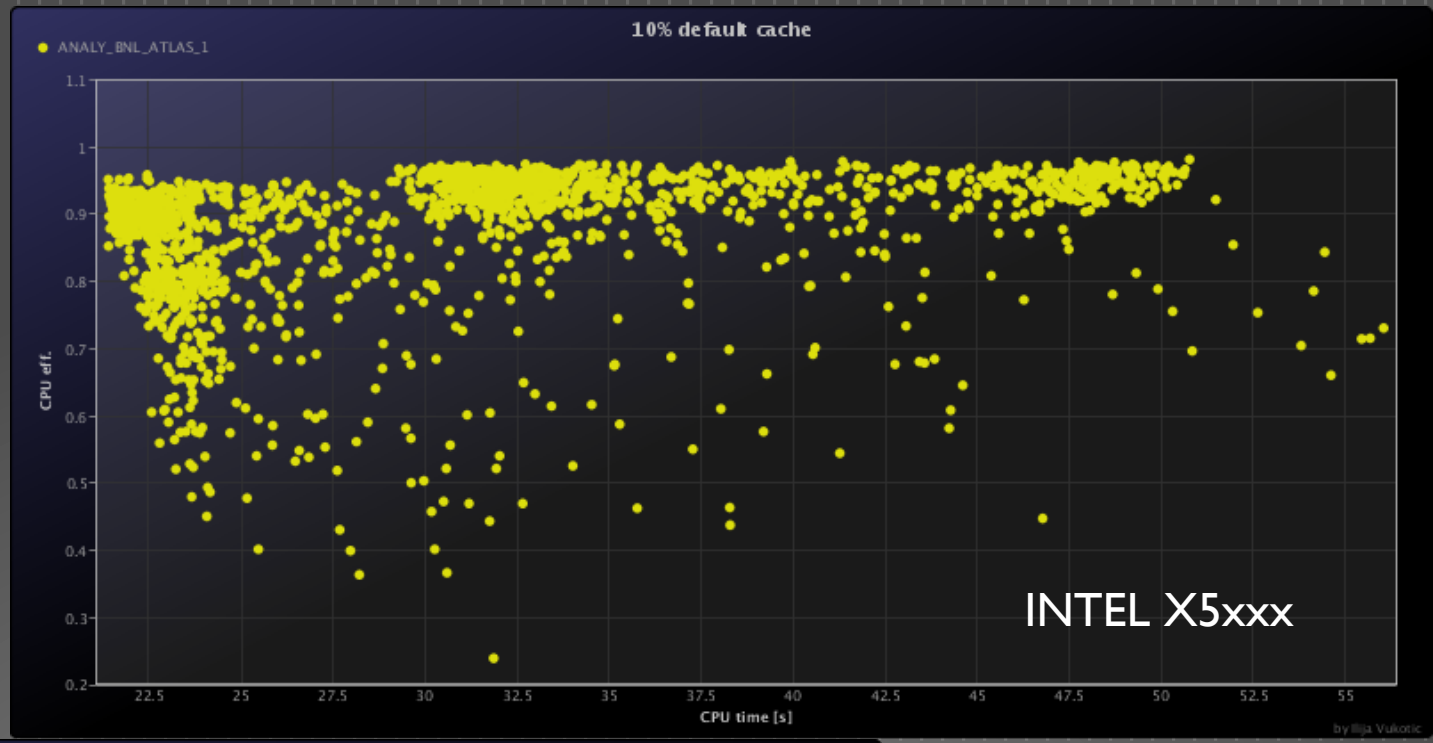
US sites

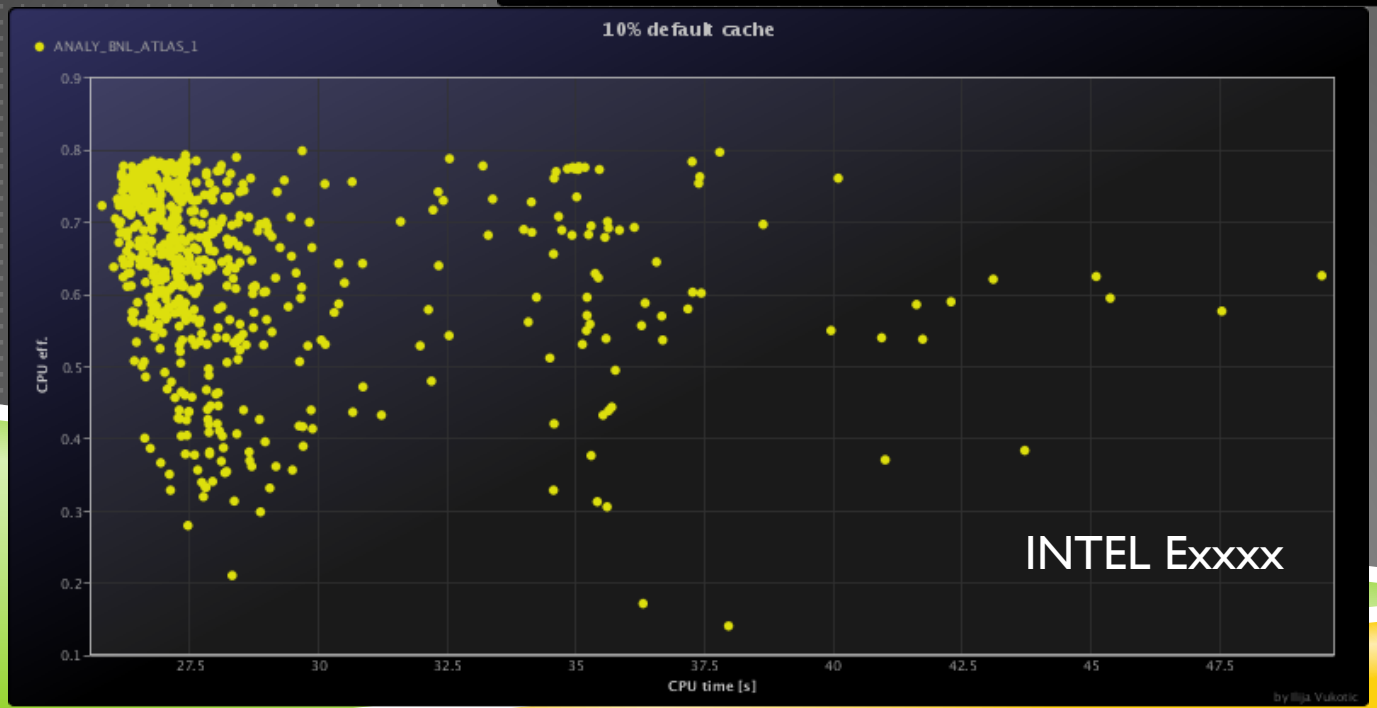






CURIOUS BNL MACHINES





US sites